REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-01-

0177

1. REPORT DATE (DD–MM–YYYY)	2. REPORT TYPE	3. DATES COVERED (From — To)
01–09–2000	Final	1 April 1997 — 30 August 2000

4. TITLE AND SUBTITLE

Mathematical Theory of Neural Networks

5a. CONTRACT NUMBER

F49620-97-1-0159

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

61102F

6. AUTHOR(S)

Eduardo D. Sontag and Héctor J. Sussmann

5d. PROJECT NUMBER

2304

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Department of Mathematics
Rutgers, The State University
Piscataway, NJ 08854-8019

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR/NM, Building 410
Bolling AFB, DC 20332-6448

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the authors and should not be construed as an official U.S. Government position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

This Final Report summarizes accomplishments of the grant research.
The work pursued under this grant dealt with artificial neural networks and other discrete/continuous models.
New bounds were obtained for sample complexity for identification of static and dynamic concept classes defined by static and recurrent networks. Structural and system-theoretic properties were characterized, leading to effective tests for identifiability and other properties.
Related models of hybrid systems were also studied; an equivalence problem for PL systems was shown to be decidable in polynomial time, and a general Maximum Principle was established for hybrid systems.

15. SUBJECT TERMS

neural networks, hybrid systems, control

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Marc Jacobs
U	U	U	UU	20	19b. TELEPHONE NUMBER (include area code) (202) 767-5027

MATHEMATICAL THEORY OF NEURAL NETWORKS

F49620-97-1-0159, Final Technical Report, 4/1/97-9/30/00

Eduardo D. Sontag and Héctor J. Sussmann
SYCON – Rutgers Center for Systems and Control
Department of Mathematics, Rutgers University
New Brunswick, NJ 08903

ABSTRACT

The work pursued under this grant dealt with artificial neural networks and other discrete/continuous models. New bounds were obtained for sample complexity for identification of static and dynamic concept classes defined by static and recurrent networks. Structural and system-theoretic properties were characterized, leading to effective tests for identifiability and other properties. Related models of hybrid systems were also studied; an equivalence problem for PL systems was shown to be decidable in polynomial time, and a general Maximum Principle was established for hybrid systems.

1 Introduction

The work pursued under this grant was centered on artificial neural networks and other discrete/continuous models for computation and systems.

For neural networks, we focused on foundational theoretical results, in the light of which algorithms used in applications (such as adaptive control, pattern recognition, of fault detection) can be compared and evaluated. For instance, our work on Vapnik-Chervonenkis dimension allows a precise quantification of the amount of data needed in order to realiably generalize from samples, in a learning or adaptive control application, and our work on identifiability permits an understanding of multiple minima in cost functions associated to numerical fitting. We also continued our work on system anad control theoretic questions associated to systems obtained by combining "saturation" sigmoidal devices which interconnect to other such devices via excitatory and inhibitory links. Towards the latter part of the grant period, we turned our attention to "spiking" neuronal models and their signal processing capabilities, as well as to the limitations imposed by noise on the computational capabilities of networks.

We also studied other hybrid models of systems and computation as part of this project. This area, broadly speaking, deals toth the interface between continuous and discrete devices (such as digital computers) used in symbolic processing. In this context, we continued the development of tools for piecewise-linear analysis, and we obtained a far-reaching generalization of the Maximum Principle of optimal control which applies to "hybrid" dynamics.

In this report, we present some of the accomplishments of the project, selected to highlight the variety of projects pursued. Complete details on this and other work done under this grant can be found in the following Web pages:

http://www.math.rutgers.edu/~sontag

http://www.math.rutgers.edu/~sussmann

2 Recurrent Neural Networks

It is said that saturation is the most commonly encountered nonlinearity in control engineering, so the development of techniques for the modeling and control of such systems is obviously of great interest. Saturations might occur in controls (discussed later) or in the rates of change of state variables. For linear systems, one is then led to the study of what are sometimes called *recurrent neural networks*, i.e. systems of the form

$$\dot{x} = \sigma(Ax + Bu)$$

or their corresponding discrete-time versions, where A and B are as usual in linear systems theory, combined with an output map $y = Cx$, typically a coordinate projection. (If we had $\sigma =$ the identity function, we would be studying continuous-time time-invariant linear systems, but typically σ is a bounded map whose translates and dilations – just as with wavelet generators – provide dense sets in appropriate function spaces, for instance tanh.) A different motivation for the study of these systems is that they arise as a very stylized model of dynamically evolving biological networks (one interprets the vector equations for x as representing the evolution of an ensemble of n "neurons," where each coordinate x_i of x is a real-valued variable which represents the internal state of the ith neuron, and each coordinate $u_i, i = 1, \ldots, m$ of u is

an external input signal; the coefficients A_{ij}, B_{ij} denote the weights, intensities, or "synaptic strengths," of the various connections, and the coordinates of $y(t) = Cx(t)$ represent the output of p probes, or measurement devices, each of which averages the activation values of several neurons). Sometimes one considers small variants of the model shown above, including for instance a linear term outside of the saturation, used to insure stability, as in the well-know model proposed by Hopfield for associative memory storage and retrieval.

Among many non-control applications, recurrent nets have been employed in the design of control laws for robotic manipulators, speech recognition, speaker identification, formal language inference, and sequence extrapolation for time series prediction. In control recurrent nets have been proposed as generic identification models or as prototype dynamic controllers, though other architectures are also used. In addition, theoretical results about neural networks established their universality as models for systems approximation as well as analog computing devices (we reported on our work along these lines in a previous grant period; see an article describing this work in *Science*, April 28, 1995). Special purpose chips have been built to implement recurrent nets directly in hardware; for instance, Hitachi's Wafer Scale Integration chips implement Hopfield nets with over 500 neurons and 30,000 synaptic connections. Electrical circuit implementations of recurrent nets, employing resistively connected networks of nonlinear amplifiers, with the resistor characteristics used to reflect the desired weights, have been suggested as analog computers, in particular for solving constrained optimization problems and for implementing content-addressable memories.

The PI started a few years ago a program of study directed to questions of controllability, stabilization, and (when outputs $y = Cx$ are considered) observability and parameter estimation for such systems. Surprisingly, explicit necessary and sufficient tests are available for observability and parameter identification, the nonlinear character (and universality) properties of the class of systems notwithstanding. We cannot provide a reasonable discussion within the constraints of this proposal, so the survey paper [17] (available in preprint form from the PI's web site) should be consulted for a recent exposition of our work in the above areas; we will limit ourselves here to describing just one area.

The paper [4] showed that the system $\dot{x} = \sigma(Ax + Bu)$ is completely controllable provided that (1) σ belongs to a certain class of nonlinearities characterized among other properties by exponential asymptotics (this class includes tanh, the typical saturation nonlinearity studied in neural networks, but it excludes arctan, which qualitatively shares boundedness, monotonicity, and concavity properties with tanh), and (2) that $B \in \mathcal{B}_{n,m}$, the $n \times m$ matrices whose rows are nonzero and distinct up to signs. An exposition can be also found in E. Sontag's textbook *Mathematical Control Theory* (second edition, Springer-Verlag, 1998). This left open a large number of questions, all of which are of great interest, foremost among them: what can be said if the hypothesis that $B \in \mathcal{B}_{n,m}$ is dropped? In general, obtaining necessary and sufficient conditions for controllability when $B \notin \mathcal{B}_{n,m}$ appears to be a very difficult subject. In [9], we showed that $B \in \mathcal{B}_{n,m}$ is necessary for a *stronger* form of complete controllability (local-local), but it is easy to see that this condition is not necessary for plain controllability. We did produce in that paper a complete solution for two-dimensional single-input ($n = 2$, $m = 1$) systems; let us summarize those results. When $B = (b_1, b_2)' \notin \mathcal{B}_{n,1}$, we may assume after a rescaling of inputs, changes of variables $x \to -x$ or $y \to -y$, and/or exchanges of variables, that one of these cases holds: $B = (0,0)'$, $B = (0,1)'$, or $B = (1,1)'$, and in the first case we don't have controllability. In the remaining two cases, under a further feedback transformation of the type $u \to ax + by + u'$, where u' is a new control, one may transform a recurrent net, while preserving controllability properties, into one of the two canonical forms: $\dot{x} = \sigma(ax + by)$, $\dot{y} = \sigma(u)$, which

are shown to be controllable if and only if $|a| \leq |b|$ and $b \neq 0$, or $\dot{x} = \sigma(ax+u)$, $\dot{y} = \sigma(by+u)$, which are controllable if and only if $a \neq b$. Obtaining a condition without dimension constraints is the final goal, but even characterizing the three-dimensional case seems nontrivial. An easier question might be to determine whether the set of pairs (A, B) which result in controllable systems (for $\sigma = \tanh$, let us say) is a semialgebraic set, as in the two-dimensional case. In addition, the case when σ does not satisfy the axioms in [4] represents an even more challenging task. These questions remain open for further research.

3 Systems with Input Saturations

One of the interesting, and somewhat unexpected, places in which neural network (feedforward sigmoidal) models have appeared is in the design of global stabilizing feedback for systems with input constraints. Often control systems are designed based on linear systems theory, which ignores amplitude limitations on inputs. However, energy, mechanical, or safety requirements often impose limits on control authority, which may result in instabilities or in undesired invariant sets (limit cycles, parasitic equilibria, etc). The classical approach to dealing with this problem has been to attempt to prevent saturation, forcing the regulated system to say within a region of linear behavior. Most "anti-windup" methods fall in this category. On the other hand, it is possible to approach the problem differently, and view the object to be controlled as a nonlinear system of the type $\dot{x} = Ax + B\sigma(u)$ where A and B are as usual in linear control theory and σ is a saturation such as tanh or the standard clipping saturation, applied coordinatewise. One line of work by the PIs deals with the study of such systems.

Fuller showed in the early 1970s that it is in general impossible to globally stabilize the origin of these systems by means of linear feedback $u{=}Fx$ even if the system is open-loop globally controllable to the origin. This suggests the obvious question of searching for *non*linear feedback laws $u{=}k(x)$ that achieve such stabilization, and in particular for nicely behaved and easily implementable controllers (in contrast to optimal control techniques, which result in highly irregular feedback). In a well-known 1990 paper, the PIs we proved that *smooth* stabilization is always possible. Motivated by our paper, soon thereafter Teel made the groundbreaking discovery that single-input multiple integrators can be stabilized by feedbacks which are themselves compositions of linear functions and iterated saturations ("nested saturation" technique). This, in turn, made us redirect our efforts to the use of Teel's technique as well as a variant (parallel saturations, which is a "neural network" architecture) in the general case of open-loop asymptotically controllable linear systems with no exponential instabilities (the rank of $[\lambda I - A, B]$ is n for all λ in the imaginary axis, and A has no eigenvalues with positive real part), obtaining general results, which appeared in various improved versions in the periods covered by the previous grant. In this grant period, we continued this study, producing a discrete-time version as well ([2]).

4 Learning Theory and Identification

The study of neural nets, and in particular of their "learning" (adaptive control, identification) capabilities, motivated us to initiate a program of research in computational learning theory, an active area of theoretical computer science. In particular, we have focused on the estimation of learning-theoretic (VC, Pollard) dimensions which are used as measures of interpolation and extrapolation ("generalization") and pattern classification power; the many publications in our

web site can be consulted for details on many projects. Our contributions in this area have been recognized by that community; for instance we have given two plenaries at NIPS, the pre-eminent and highly selective conference in the area, and were asked to deliver a short course on neural network learning at a Newton Institute summer program (lectures described in [18]).

One recent direction of study has been the generalization of dimension estimates for linear systems obtained in a previous grant period (*IEEE Trans. Inform. Theory* **42** (1996): 1479-1487) to discrete ([7]) and continuous ([8]) time nonlinear systems, and especially the study of dimension estimates, and their implications for sample complexity of worst case identification for linear systems subject to bandwidth-restricted inputs ([19], [15]). That work takes a computational learning theory approach to a problem of linear systems identification. It is assumed there that input signals have only a finite number k of frequency components, and systems to be identified have dimension no greater than n. The main result established that the sample complexity needed for identification scales polynomially with n and logarithmically with k. Let us provide some details of this particular work.

4.1 Learning and Linear Systems Identification

The problem of systems identification may be seen as an instance of the general question of "learning" an unknown function. Techniques from Computational Learning Theory (CLT) can be applied and our previous papers (previous grant period) had already provided results applicable to the identification of discrete-time linear systems on finite-window data. For continuous-time systems, the situation is complicated by the fact that, even for finite-length inputs, learnability is impossible when formulated in the CLT framework, as can be seen by applying the discrete-time results (through sampling). Thus, in our work, we supposed that all inputs to be used, in the learning as well as in validation stages, belong to the linear span of a fixed number k of sinusoidal basic functions. This band-limiting assumption allowed us to obtain a precise result: the sample complexity needed for identification scales polynomially on an upper bound on the systems being identified, and logarithmically with k. This provides a tight analogy to the discrete results previously obtained, in which k appeared as the length of the discrete-time window employed.

In the context of learning we discuss continuous-time linear control systems:

$$\dot{x} = Ax + Bu, \qquad x(0) = x^0, \qquad y = Cx, \tag{1}$$

where A, B, and C are $n \times n$, $n \times m$, and $p \times n$ real matrices, and the time interval is [0,1]. We study sign-observations

$$\text{sign } y(1) = (\text{sign } y_1(1), \ldots, \text{sign } y_p(1))^T,$$

where sign $z = 0$, if $z \leq 0$, sign $z = 1$, if $z > 0$ and ^{T} stands for the transpose. For scalar observations this is a classification problem; each output is classified either 0 or 1 and the VC-dimension can be used to study the learning complexity of the problem. (When $p > 1$, a generalization of the VC-dimension or a loss function is needed.)

We consider controls $u = (u_1, \ldots, u_m)$ such that

$$u = G\omega,$$

where G is a $m \times k$ matrix that parametrizes the control. The set of basis input functions $\Omega = \{\omega_1, \ldots, \omega_k\}$ is fixed. The bounds for the VC-dimension or other complexity dimensions

will depend on the properties of the set Ω. For scalar inputs (i.e., $m = 1$) the VC-dimension associated to the mapping from inputs G to scalar sign-observations is bounded by k, which in fact can be very large in applications. This bound is tight; we give an example of a function class Ω for which the associated VC-dimension is indeed k. By considering band-limited controls the bound can be improved. In this work we consider the following set of basis input functions

$$\Omega = \Big\{\omega_1, \ldots, \omega_k \; ; \qquad \omega_1, \ldots, \omega_k \;\text{ linearly independent and}$$
$$\omega_j = t^{\ell_j} e^{\alpha_j t} \sin(\beta_j t) \;\text{ or }\; \omega_j = t^{\ell_j} e^{\alpha_j t} \cos(\beta_j t)$$
$$\text{with }\; \ell_j \in \mathbb{N}, \alpha_j, \beta_j \in \mathbb{R}, j = 1, \ldots, k\Big\},$$

and let

$$\ell_{\max} = \max\{\ell_1, \ldots, \ell_k\}. \tag{2}$$

Order the set of basis input functions Ω and denote $\omega = (\omega_1, \ldots, \omega_k)^T$. Let

$$X_\Omega = \{G\omega : [0,1] \to \mathbb{R}^m \; ; \; G \in \mathbb{R}^{mk}\},$$

and for each linear system $\Sigma = (A, B, C, x^0)$ of dimension n define the mapping $\Phi_\Sigma : X_\Omega \to \mathbb{R}^p$ by $\Phi_\Sigma(G\omega) = y(1)$, where $y(1)$ is the solution of Σ with control $u = G\omega$. Similarly we define the mapping for sign-observations,

$$S_\Sigma : X_\Omega \to \{0,1\}^p \quad G\omega \mapsto \operatorname{sign}(\Phi_\Sigma(G\omega)).$$

The class of above mappings is the *sign system concept class*

$$\mathcal{C}_{m,p} = \{S_\Sigma \; ; \; \Sigma \text{ linear system of dimension } \; n\}.$$

Theorem [Sample complexity for concept learning]. For sign systems concept class $\mathcal{C}_{m,1}$ with scalar observations, i.e., $p = 1$, the sample complexity $s(\epsilon, \delta)$ for identifiers that agree with the observed sample can be bounded as

$$s(\epsilon, \delta) \leq \max\left\{\frac{8 \text{ VC } (\mathcal{C}_{m,1})}{\epsilon} \log_2\left(\frac{8e}{\epsilon}\right), \frac{4}{\epsilon}\log_2\left(\frac{2}{\delta}\right)\right\},$$

where

$$\text{VC } (\mathcal{C}_{m,1}) \leq 2(2mn^2 + 4n + 1)\log_2\Big[8e(8mn^2k(n + \ell_{\max}) + 1)(2nk + 2(1+2k)^n)\Big]$$

and $\ell_{\max}$ is given by (2).

In terms of n (the dimension of the state space) and k (the band-width) the upper bound for the VC-dimension is of the form $O(n^3 \log_2(nk))$. We provided also VC-dimension lower bound, which is, in terms of the band-width, of the form $O(\log(k))$. In particular, in a typical setting of fairly small system dimension n and large band-width k, the $\log k$ bound is a clear improvement over the linear bound given by elementary analysis.

In our work, we illustrated how the system (1) with $x(0) = 0$ can be parametrized by $n(m+1)$ parameters. In the following definition we take the final time to be $\tau > 1$ in order to show the effect of the time interval in the learning complexity:

Let $\lambda \in \mathbb{R}^{n(m+1)}$ be the system parameters as above with $\|\lambda\|_\infty = \max_{1 \leq i \leq n(m+1)} \lambda_i < 1$ and let $F(\lambda, u) = y(\tau)$ be the solution of (1) with system parameters λ and control $u =$

$(u_1, \ldots, u_m) \in U = \{u = (u_1, \ldots, u_m)\ ;\ \int_0^\tau u_i(t)dt \leq M,\ i = 1, \ldots, m\}$. The *class with bounded controls* is defined as

$$\mathcal{F}_B = \{F(\lambda, \cdot): U \to \mathbb{R}\ ;\ \|\lambda\|_\infty < 1\}.$$

Theorem[Sample complexity for proper agnostic learning]. Let $\kappa > 0$, then the class $\mathcal{F}_B$ is properly agnostically learnable from

$$O\left(\frac{1}{\epsilon^2}\left(\text{ fat }_{(1/4-\kappa)\epsilon}(\mathcal{F}_B)\log^2\frac{1}{\epsilon} + \log\frac{1}{\delta}\right)\right)$$

samples, where

$$\text{fat }_{(1/4-\kappa)\epsilon}(\mathcal{F}_B) \leq \min\left\{\begin{array}{l}(m+1)n\log_2\left\lfloor\frac{n^2 m\tau^n e^\tau kM}{(1/4-\kappa)\epsilon}\right\rfloor, \\ 2(m+4)n\log_2\Big(8e(nmk4(n+\ell_{\max})+1)(2nk+2(2k+1)^n)\Big),\end{array}\right.$$

together with $\ell_{\max}$ given by (2) and (2), and M a constant satisfying

$$\int_0^\tau |u_i(\tau - t)|dt \leq kM$$

for all $i = 1, \ldots, m$. In above, $\lfloor x \rfloor$ stands for the integer part of x.

Let us discuss briefly the techniques used. When the basis input functions $\omega_1, \ldots, \omega_k$ satisfy certain rationality condition associated to the control system (we split the rational function into pieces without poles) we show that the sign of the final state can by computed by a Boolean formula evaluating polynomial equalities and inequalities. Then the complexity bound can be obtained by counting arguments and using a result by Goldberg and Jerrum (1995). We prove lower bounds for the VC-dimension with scalar sign-observations. In comparison to the upper bound, the lower bound is more general; we just need to assume that the basis input functions are continuous and independent. The bound is proved by using dual VC-dimension and axis shattering introduced in our previous work. The bounds on the fat-shattering dimension associated with proper agnostic learning are obtained with a very simple technique. The paper contains also pseudo-dimension bounds with respect to loss functions that preserve the rationality structure of the output.

5 Piecewise-Linear ("Hybrid") Systems

Artificial neural networks are sometimes proposed as a framework in which to integrate symbolic and numeric computation (a point of view emphasized in and an alternative source of models for nonlinear control and identification. A different but parallel avenue to some of the same conceptual issues is provided by the area now known as "hybrid systems" theory. Hybrid systems theory has recently become the focus of increased research, as evidenced for instance by the many conferences and workshops in the area. The PI is recognized as having originated one of the first approaches to hybrid systems analysis, the theory of discrete-time *piecewise linear systems* (PLS) introduced in the early 1980s (*IEEE Trans. Autom. Control* **26**(1981): 346-358.) Recently, several teams have initiated other research efforts on PLS. For instance, Morari and his group at the ETH showed recently that the general class of hybrid "Mixed Logical Dynamical (MLD)" systems is in a precise sense equivalent to that of PLS as introduced by the PI, and based on this equivalence, and using tools from piecewise affine systems, studied basic system-theoretic properties and suggested numerical tests based on mixed-integer linear

programming for checking controllability and observability. Recently, we were able to prove the polynomial-time solvability of the state equivalence problem, which was a long-standing open question, see [11].

Among the most basic questions which can be asked about any class of systems are those regarding equivalence, such as: given two systems, do they represent the same dynamics under a change of variables? As a preliminary step in answering such a question, one must determine if the state spaces of both systems are isomorphic in an appropriate sense. That is, one needs to know if an invertible change of variables is at all possible. Only later can one ask if the equations are the same. For classical, finite dimensional linear systems, this question is trivial, since only dimensions must match. For finite automata, similarly, the question is also trivial, because the cardinality of the state set is the only property that determines the existence of a relabeling of variables. For other classes of systems, however, the question is not as trivial, and single numbers such as dimensions or cardinalities may not suffice to settle the equivalence problem in the respective category. Given that the class of behaviors that can be represented by PLS is extremely large, it should come as no surprise that many of the basic verification and design objectives are NP-hard or even undecidable, as we have remarked in various publications. In our orignal work (*Pacific J.Math.*, **98**(1982): 183-201), we provided a characterization of the Grothendieck group of the category, as well as a generalization of the Euler characteristic for polyhedra (and certain theorems for Euler characteristics become trivial when interpreted in these terms). Moreover, we proved existence of an algorithm for deciding if two PL sets (given in terms of formulas in L) are isomorphic, via results on decidability of word problems and results of Eilenberg and Schützenberger on finitely generated commutative monoids. Thus the isomorphism problem is one problem that *is* decidable. However, the algorithm that results from that approach has exponential time complexity. Obviously, having a polynomial time algorithm should have a major impact on future studies of PL systems.

5.1 Some more details

In order to sketch the basic definitions for PL algebra and PL systems, it is convenient to introduce the first order theory of the real numbers with addition and order. That is, we take the first-order language L consisting of constants r and unary functions symbols $r(\cdot)$, for each real number r (the latter corresponding to "multiplication by the constant r"), as well as binary function symbol $+$ and relation symbols $>$ and $=$. A basic fact is that a quantifier elimination theorem holds: *every set defined by a formula in L is a PL set.* That is to say, for any formula $\Phi(x)$ with n free variables $x = x_1, \ldots, x_n$, the set $\{x \mid \Phi(x)\}$ is a PL set. (Of course, we can enlarge the language by adding symbols for sets and maps already known to be PL.) This fact is very simple to establish and it provides a very convenient tool for establishing the basic theoretical properties of PL systems. Moreover, the proofs of these facts are constructive, in that the actual quantifier algorithm could be in principle used to compute feedback laws and the like. Another constructively-proved fact from our 1982 paper is the following "global implicit function theorem": Assume that $\phi : X \times Y \to \mathbb{R}^n$ is a PL map, and assume that for each x the equation $\phi(x, y) = 0$ can be solved for y. Then there is a PL map $\pi : X \to Y$ so that $\phi(x, \pi(x)) = 0$ for all x. (Equivalently: for any PL subset $R \subseteq X \times Y$ with onto projection into X, there is a PL map $\pi : X \to Y$ (a "section") so that $(x, \phi(x)) \in R$ for all $x \in X$.) This fact is central to the existence of feedback controllers.

A PL isomorphism is nothing else than an operation of the following type: make a finite number of cuts along a set of lines (or segments), apply an affine (linear plus translation)

transformation to each piece (not dropping any lower-dimensional pieces), and finally paste it all together. As an example, let us take the interior of the triangle in $\mathbb{R}^2$ obtained as $\mathrm{oc}\{(0,0),(1,1),(2,0)\}$, where we are using "oc" to indicate the interior of the convex hull of the corresponding points. (We can also define this set, of course, as the intersection of the three hyperplanes $x_2 > 0$, $x_1 - x_2 > 0$, and $x_1 + x_2 < 2$.) We now show that this triangle is PL isomorphic to the interior of the open square with vertices $(0,0)$, $(1,1)$, $(0,1)$, and $(1,0)$. First we cut along the segment $S_1 = \mathrm{oc}\{(1,0),(1,1)\}$, obtaining the union of S_1, S_2, and S_3, where $S_2 = \mathrm{oc}\{(0,0),(1,0),(1,1)\}$ and $S_3 = \mathrm{oc}\{(1,1),(1,0),(2,0)\}$. Next, we apply the affine transformation

$$Tx = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} x - \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

to change S_3 into $S_3' = \mathrm{oc}\{(1,1),(0,0),(0,1)\}$. Finally, we apply the affine transformation

$$Tx = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} x - \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

to change S_1 into the missing diagonal $S_1' = \mathrm{oc}\{(0,0),(1,1)\}$, and we glue it all back. See Figure 1.

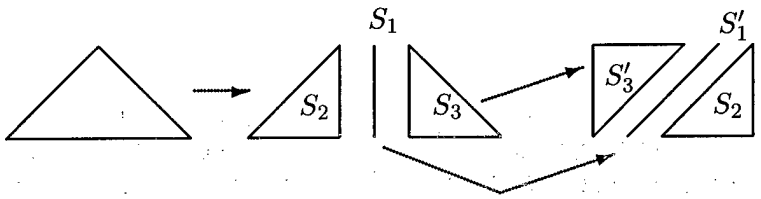

Figure 1: Example: triangle is PL isomorphic to square

One of the main results in our early 1980s work on piecewise-linear algebra provided a classification of PL sets under isomorphism. The critical step in this classification is to associate to each PL set X a "label" with the property that two spaces X and Y are isomorphic if and only if their labels are related in a certain manner. (By analogy, two finite-dimensional real vector spaces are linearly isomorphic if and only if their dimensions are the same, i.e., letting the "label" be the dimension, if their labels coincide. But in the PL case, single integers do not suffice as "labels".) Labels are, by definition, polynomials in two variables x, y with non-negative integer coefficients. We let $\mathbb{N}[x,y]$ denote the collection of all such polynomials. Examples of labels are 1, x, y, x^3, $1+xy+x^2$, etc. We interpret the sum in $\mathbb{N}[x,y]$ as union of disjoint sets and the product as Cartesian product of sets, the unit 1 as a one-element set, the variable x as the open interval $(0,1)$, and the variable y as the half-line $(0,+\infty)$. Thus, x^3 is an open cube, and $1+xy+x^2$ is the union of a point, a disjoint set $(0,1)\times(0,+\infty)$, and a unit square disjoint from both. One may decompose any PL set into a finite union (algebraically, a sum) of objects each of which is linearly isomorphic to a monomial in x and y. (Simplicial decompositions provide a way to do this.) In this manner, a label (nonunique) can be associated to each PL set.

Certain formal equalities are easy to establish. Splitting the interval x as

$$(0,1) = (0,1/2) \bigcup \{1/2\} \bigcup (1/2,1),$$

and then using affine maps ($t \mapsto 2t$ and $t \to 2t-1$ respectively) to map the first and last interval to x, we obtain "$x = 2x+1$". On the other hand, the split $y = (0,+\infty) = (0,1) \bigcup \{1\} \bigcup (1,+\infty)$ (and $t \mapsto t-1$ applied to the last set) gives us the identity "$y = x + 1 + y$". Drawing a bisecting line through the first quadrant in $\mathbb{R}^2$ gives "$y^2 = y^2 + y + y^2$" (using, e.g., the linear transformation $(t_1, t_2) \mapsto (t_1 - t_2, t_2)$ to send the lower triangle $\{(t_1, t_2) | t_1 > 0, t_1 > t_2\}$ to y^2).

It was shown in our previous work that these three identities are enough, in the sense that two sets are isomorphic if and only if their labels can be obtained from each other by using repeatedly these elementary identities. In other words, isomorphism is precisely determined by the congruence generated by these identities in the semiring $\mathbb{N}[x, y]$. In this manner, one may apply to the equivalence problem the results of Eilenberg and Schützenberger on finitely generated commutative monoids that are obtainedd by quotients under such congruences. Equivalence under congruences is in general non-polynomial time; however, exploiting the special form of the congruences that define PL equivakence, we were able in [11] to find a polynomial time algorithm for our problem. Our collaborator on this project, B. Dasgupta, has recently supervised a Master's thesis implementing the algorithm. As mentioned earlier, this is only a first step in studying equivalence of PL systems, and further work is ongoing.

6 Networks of Spiking Neurons

We have also continued work on a different type of network which represents neural populations, based on "spiking neurons" (information is encoded in inter-spike intervals). This biologically more realistic and appealing class of systems gives rise to a whole new set of questions. Some of our results regarding such models are outlined next. (Let us just add here that we have also made initial progress towards the characterization of the structure of local minima of associated fitting problems, using a combination of differential topology (Morse theoretic), logic, and algebraic-geometric techniques such as previously employed in our work on critical points of objective functions involving sigmoidal networks (*Advances in Computational Mathematics*, **5**(1996): 245-268) and the geometry of Banach space techniques from our paper [1], in the count of minima and the study of approximation rates.)

Experimental data show that biological synapses behave quite differently from the symbolic synapses in all common artificial neural network models. Biological synapses are dynamic, i.e., their "weight" changes on a short time scale by several hundred percent in dependence of the past input to the synapse. In [12], we addressed the question how this inherent synaptic dynamics – which should not be confused with long term "learning" – affects the computational power of a neural network. In particular we analyzed computations on temporal and spatio-temporal patterns, and we gave a complete mathematical characterization of all filters that can be approximated by feedforward neural networks with dynamic synapses. It turns out that even with just a single hidden layer such networks can approximate a very rich class of nonlinear filters: all filters that can be characterized by Volterra series. This result is robust with regard to various changes in the model for synaptic dynamics. Our characterization result provideed for all nonlinear filters that are approximable by Volterra series a new complexity hierarchy which is related to the cost of implementing such filters in neural systems. This set of results has given rise to several follow-up papers, and has attracted considerable attention in the theoretical neuroscience community. Let us give some details next.

Synapses in common artificial neural network models are static: the value w_i of a synaptic weight is assumed to change only during "learning". In contrast to that, the "weight" $w_i(t)$ of

a biological synapse at time t is known to be strongly dependent on the inputs $x_i(t-\tau)$ that this synapse has received from the presynaptic neuron i at previous time steps $t-\tau$. Several recent papers have shown that a model of the form

$$w_i(t) = w_i \cdot D(t) \cdot (1 + F(t)) \tag{3}$$

with a constant w_i, a depression term $D(t)$ with values in $(0, 1]$, and a facilitation term $F(t) \geq 0$, can be fitted remarkably well to experimental data for synaptic dynamics. The facilitation term $F(t)$ is usually modeled as a linear filter with exponential decay: If $x_i(t-\tau)$ is the output of the presynaptic neuron (typically modeled by a sum of δ-functions), then the current value of this facilitation term is of the form

$$F(t) \;=\; \rho \int_0^\infty x_i(t-\tau) \cdot e^{-\tau/\gamma} d\tau \tag{4}$$

for certain parameters $\rho, \gamma > 0$ that vary from synapse to synapse. The analysis in our work is primarily based on this model, but we also showed that our results also hold for the somewhat more complex models for synaptic dynamics obtained in a mean-field context.

We showed in [12] that such inherent synaptic dynamics empowers neural networks with a remarkable capability for carrying out computations on temporal patterns (i.e., time series) and spatio-temporal patterns. This computational mode, where inputs and outputs consist of temporal patterns or spatio-temporal patterns – rather than static vectors of numbers – appears to provide a more adequate framework for analyzing computations in biological neural systems. Furthermore their capability for processing temporal and spatio-temporal patterns in a very efficient manner may be linked to their superior capabilities for real-time processing of sensory input, hence our analysis may provide new ideas for designing artificial neural systems with similar capabilities.

We considered not just computations of neural systems with a *single* temporal pattern as input, but also characterize their computational power for the case where *several* different temporal patterns $u_1(t), \ldots, u_n(t)$ are presented in parallel as input to the neural system. Hence we also provided a complete characterization of the computational power of feedforward neural systems for the case where salient information is encoded in temporal correlations of firing activity in different pools of neurons (represented by correlations among the corresponding continuous functions $u_1(t), \ldots, u_n(t)$). Therefore various informal suggestions for computational uses of such code can be placed on a rigorous mathematical foundation: It is easy to see that a large variety of computational operations that respond in a particular manner to correlations in temporal input patterns define time invariant filters with fading memory, hence they can in principle be implemented on each of the various kinds of dynamic networks considered in our work. Previous standard models for computations on temporal patterns in artificial neural networks are time-delay neural networks (where temporal structure is transformed into spatial structure) and recurrent neural networks, both being based on standard "static" synapses. Such transformation makes it impossible to let "time represent itself" (in the language of Mead) in subsequent computations, which tends to result in a loss of computational efficiency. The results of our work suggest that feedforward neural networks with simple dynamic synapses provide an attractive alternative.

6.1 More Details

In contrast to the static output of gates in feedforward artificial neural networks, the output of biological neurons consists of action potentials ("spikes"), i.e., stereotyped events that mark

certain points in time. These spikes are transmitted by synapses to other neurons, where they cause changes in the membrane potential that affect the times when these other neurons fire and thereby emit a spike. Empirical data describes the amplitudes of EPSC's (excitatory postsynaptic currents) in a neuron in response to a spike train from a presynaptic neuron. These two neurons are likely to be connected by multiple synapses, and the resulting EPSC amplitude can be understood as a population response of these multiple synapses. Therefore it is justified to employ a deterministic model for synaptic dynamics in spite of the stochastic nature of synaptic transmission at a single release sit. The EPSC amplitude in response to a spike is modeled by terms of the form $w \cdot (1 + \mathcal{F})$ and $w \cdot \mathcal{D} \cdot (1 + \mathcal{F})$, where $\mathcal{F}$ is a linear filter with impulse response $\rho \cdot e^{-\tau/\gamma}$ modeling facilitation and D is some nonlinear filter modeling depression at synapses. In some versions of the model considered in the literature, this filter D consists of several depression terms. However it only assumes values > 0 and is always time invariant and has fading memory.

We analyzed the impact of this synaptic dynamics in the context of common models for computations in populations of neurons where one can ignore the stochastic aspects of computation in individual neurons in favor of the deterministic response of pools of neurons that receive similar input ("population coding" or "space rate coding"). More precisely, we based our neural network model is based on a mean-field analysis of networks of biological neurons, where pools P of neurons serve as computational units, whose time-varying firing activity (measured as the number of neurons in P that fire during a short time interval $[t, t + \Delta]$) is represented by a continuous bounded function $y(t)$. In case that pool P receives inputs from m other pools of neurons $P_1, \ldots, P_m$, we assume that $y(t) = \sigma(\sum_{i=1}^{m} w_i(t)x_i(t) + w_0)$, where $x_i(t)$ represents the time-varying firing activity in pool P_i and $w_i(t)$ represents the time-varying average "weight" of the synapses from neurons in pool P_i to neurons in pool P. (The function $\sigma : \mathbb{R} \to \mathbb{R}$ is some "activation function", for example $\sigma(x) = 1/(1 + e^{-x})$; for the theorems, it suffices to assume that σ is continuous and not a polynomial.) We allow a general representation of the dynamics of synapses from a nonlinear filter applied to a sequence of δ-functions (i.e., to a spike train) to be a nonlinear filter applied to a continuous input function $x_i(t)$. Thus, if $x_i(t)$ is a continuous function describing the firing activity in the ith presynaptic pool P_i of neurons we model the size of the resulting synaptic input to a subsequent pool P of neurons by terms of the form $w_i(t) \cdot x_i(t)$ with $w_i(t) := w_i \cdot (1 + \mathcal{F}x_i(t))$ or $w_i(t) := w_i \cdot \mathcal{D}x_i(t) \cdot (1 + \mathcal{F}x_i(t))$, where the filters $\mathcal{F}$ and D are defined as in previous literature. The first equation that just models facilitation gives rise to the definition of the class DN of dynamic networks, and the second equation, that models the more common co-occurrence of facilitation and depression, gives rise to the definition of the class DN*.

We define the class DN of *dynamic networks* as the class of arbitrary feedforward networks consisting of sigmoidal gates that map input functions $x_1(t), \ldots, x_m(t)$ to a function

$$y(t) = \sigma(\sum_{i=1}^{m} w_i(t)x_i(t) + w_0),$$

with

$$w_i(t) = w_i \cdot (1 + \rho \int_0^{\infty} x_i(t - \tau)e^{-\tau/\gamma} d\tau)$$

for parameters $w_i \in \mathbb{R}$ and $\rho, \gamma > 0$. σ is some "activation function" from $\mathbb{R}$ into $\mathbb{R}$, for example the logistic sigmoid function defined by $\sigma(x) = 1/(1 + e^{-x})$. We will assume in the following only that σ is continuous and not a polynomial. The slightly different class DN* is defined in

the same way, except that $w_i(t)$ is of the form

$$w_i(t) = w_i \cdot \mathcal{D}x_i(t) \cdot (1 + \rho \int_0^\infty x_i(t-\tau)e^{-\tau/\gamma}d\tau),$$

where D is some arbitrary *given* time invariant fading memory filter with values $\mathcal{D}x_i(t) \in (0,1]$. Thus dynamic networks in DN or DN* are simply feedforward neural networks consisting of sigmoidal neurons, where static weights w_i are replaced by biologically realistic history-dependent functions $w_i(t)$. The input to a dynamic network consists of an arbitrary vector of functions $u_1(\cdot), \ldots, u_n(\cdot)$. The output of a dynamic network is defined as weighted sum

$$z(t) = \sum_{i=1}^{k} \alpha_i y_i(t) + \alpha_0$$

of the time-varying outputs $y_1(t), \ldots, y_k(t)$ of certain sigmoidal neurons in the network, where the "weights" $\alpha_0, \ldots, \alpha_k$ can be assumed to be static. Thus a dynamic network with n inputs maps n input functions $u_1(\cdot), \ldots, u_n(\cdot)$ onto some output function $z(\cdot)$.

Networks that operate on temporal patterns map functions of timae onto functions of time. Let us call these operators *filters*. We will reserve the letters $\mathcal{F}, \mathcal{H}, \mathcal{S}$ for filters, and we write $\mathcal{F}\underline{u}$ for the function resulting from an application of the filter $\mathcal{F}$ to a vector $\underline{u}$ of functions. Notice that when we write $\mathcal{F}\underline{u}(t)$ we mean, of course, $(\mathcal{F}\underline{u})(t)$ (that is, the function $\mathcal{F}\underline{u}$ evaluated at time t). We write $C(A,B)$ for the class of all continuous functions $f : A \to B$. We will consider suitable subclasses $U \subseteq C(A,B)$ for $A \subseteq \mathbb{R}^k$ and $B \subseteq \mathbb{R}$, and study filters that map U^n into $\mathbb{R}^{\mathbb{R}}$ (where $\mathbb{R}^{\mathbb{R}}$ is the class of all functions from $\mathbb{R}$ into $\mathbb{R}$), i.e. filters that map n functions $u(\cdot), \ldots, u_n(\cdot)$ onto another function $z(\cdot)$. Let us focus for simplicity on the case $k = 1$, i.e. the case where the input functions $u_1(\cdot), \ldots, u_n(\cdot)$ are functions of a single variable – which we will interpret as time. The case $k > 1$ (spacio-temporal patters) was also studied in our work.

A trivial special case of a filter is the shifting filter $\mathcal{S}_{t_0}$ with $\mathcal{S}_{t_0}u(t) = u(t-t_0)$. An arbitrary filter $\mathcal{F} : U^n \to \mathbb{R}^{\mathbb{R}}$ is called *time invariant* if a shift of the input functions by a constant t_0 just causes a shift of the output function by the same constant t_0, i.e., if for any $t_0 \in \mathbb{R}$ and any $\underline{u} =\leq u_1, \ldots, u_n\rangle \in U^n$ one has that $\mathcal{F}\underline{u}_{t_0}(t) = \mathcal{F}\underline{u}(t-t_0)$ where $\underline{u}_{t_0} =\leq \mathcal{S}_{t_0}u_1, \ldots, \mathcal{S}_{t_0}u_n\rangle$. All filters considered in our work are time invariant. Note that if U is closed under $\mathcal{S}_{t_0}$ for all $t_0 \in \mathbb{R}$ then a time invariant filter $\mathcal{F} : U^n \to \mathbb{R}^{\mathbb{R}}$ is fully characterized by the values $\mathcal{F}\underline{u}(0)$ for $\underline{u} \in U^n$.

Another essential property of filters considered in our work was "fading memory" in the sense of Boyd and Chua. If a filter $\mathcal{F}$ has fading memory then the value of $\mathcal{F}\underline{v}(0)$ can be approximated arbitrarily closely by the value of $\mathcal{F}\underline{u}(0)$ for functions $\underline{u}$ that approximate the functions $\underline{v}$ for sufficiently long bounded intervals $[-T, 0]$. The formal definition is as follows: a filter $\mathcal{F} : U^n \to \mathbb{R}^{\mathbb{R}}$ has *fading memory* if for every $\underline{v} =\leq v_1, \ldots, v_n\rangle \in U^n$ and every $\varepsilon > 0$ there exist $\delta > 0$ and $T > 0$ so that $|\mathcal{F}\underline{v}(0) - \mathcal{F}\underline{u}(0)| < \varepsilon$ for all $\underline{u} =\leq u_1, \ldots, u_n\rangle \in U^n$ with the property that $\|\underline{v}(t) - \underline{u}(t)\| < \delta$ for all $t \in [-T, 0]$.

It is obvious that any filter $\mathcal{F}$ which can be represented by a sum of finitely many Volterra terms of any order (i.e., by a Volterra polynomial or finite Volterra series) is time invariant and has fading memory. This holds for any class U of uniformly bounded input functions u. Both of these properties are inherited by filters $\mathcal{F}$ that can be approximated by some arbitrary infinite sequence of such filters. This implies that any filter that can be approximated by finite or infinite Volterra series (which converge in the sense used here) is time invariant and has fading memory (over any class U of uniformly bounded functions u). Boyd and Chua showed

in 1985 that under reasonable additional assumptions about U the converse also holds: any time invariant filter $\mathcal{F} : U \to \mathbb{R}^{\mathbb{R}}$ with fading memory can be approximated arbitrarily closely by Volterra polynomials.

One of our theorems shows that simple filters that only model synaptic facilitation (as considered in the definition of DN) provide the networks already with sufficient dynamics to approximate arbitrary given time invariant filters with fading memory. We show that the simultaneous occurrence of depression (as in DN*) is not needed for that, but it also does not hurt. This appears to be of some interest for the analysis of computations in biological neural systems, since a fairly large variety of different functional roles have already been proposed for synaptic depression: explaining psychological data on conditioning and reinforcement (Grossberg), boundary formation in vision and visual persistence, switching between different neural codes, and automatic gain control. As a complementation of these conjectured roles for synaptic depression, we also proved a theorem which points to a possible functional role for synaptic facilitation: it empowers even very shallow feedforward neural systems with the capability to approximate basically any linear or nonlinear filter that appears to be of interest in a biological context. Furthermore we show that this possible functional role for facilitation can co-exist with independent other functional roles for synaptic depression: Our result shows that one can first choose the parameters that control synaptic depression to serve some other purpose, and can then still choose the parameters that control synaptic facilitation so that the resulting neural system can approximate any given time invariant filter with fading memory.

Theorem. Assume that U is the class of functions from $\mathbb{R}$ into $[B_0, B_1])$ which satisfy $|u(t) - u(s)| \leq B_2 \cdot |t - s|$ for all $t, s \in \mathbb{R}$, where B_0, B_1, B_2 are arbitrary real-valued constants with $0 < B_0 < B_1$ and $0 < B_2$. Let $\mathcal{F}$ be an arbitrary filter that maps vectors $\underline{u} =\leq u_1, \ldots, u_n\rangle \in U^n$ into functions from $\mathbb{R}$ into $\mathbb{R}$.

Then the following are equivalent:

(a) $\mathcal{F}$ can be approximated by dynamic networks $\mathcal{S} \in DN$ (i.e., for any $\varepsilon > 0$ there exists some $\mathcal{S} \in$ DN such that $|\mathcal{F}\underline{u}(t) - \mathcal{S}\underline{u}(t)| < \varepsilon$ for all $\underline{u} \in U^n$ and all $t \in \mathbb{R}$)

(b) $\mathcal{F}$ can be approximated by dynamic networks $\mathcal{S} \in DN$ with just a single layer of sigmoidal neurons

(c) $\mathcal{F}$ is time invariant and has fading memory

(d) $\mathcal{F}$ can be approximated by a sequence of (finite or infinite) Volterra series.

These equivalences remain valid if DN is replaced by DN*.

The following result follows from the above Theorem. It shows that the class of filters that can be approximated by dynamic networks is very stable with regard to changes in the definition of a dynamic network.

Corollary. Dynamic networks with just one layer of dynamic synapses and one subsequent layer of sigmoidal gates can approximate the same class of filters as dynamic networks with an arbitrary finite number of layers of dynamic synapses and sigmoidal gates. Even with a sequence of dynamic networks that have an unboundedly growing number of layers one cannot approximate more filters.

Furthermore if one restricts the synaptic dynamics in the definition of dynamic networks to the simplest form $w_i(t) = w_i \cdot (1 + \rho \int_0^\infty x_i(t - \tau)e^{-\tau/\gamma}d\tau)$ with some arbitrarily *fixed* $\rho > 0$ and

time constants γ from some arbitrarily *fixed* interval $[a, b]$ with $0 < a < b$, the resulting class of dynamic networks can still approximate (with just one layer of sigmoidal neurons) any filter that can be approximated by a sequence of arbitrary dynamic networks as defined. In the case of DN* one can either choose to fix $\rho > 0$ or one can arbitrarily fix the interval $[a, b]$ for the value of γ.

7 Optimal Control of Hybrid Systems

The problem of optimal control for hybrid systems, mixing continuous and discrete variables, is recognized as one of the central challenges in the emerging hybrid system area, and work carried out under this grant resulted in substantial advances. Indeed, the papers [20], [21], and [22] provided different versions of the *Maximum Principle* of optimal control for hybrid systems, under minimal regularity conditions. In this short summary, we will define the class of hybrid problems to be considered and then state informally the Maximum Principle, leaving aside a detailed specification of technical assumptions. (The references given above should be consulted for all details.) The results in the papers [20], [21], and [22] are stronger than the usual versions of the finite-dimensional maximum principle. For example, even the theorem for classical differentials applies to situations where the maps are not of class C^1, and can fail to be Lipschitz continuous. The "nonsmooth" result applies to maps that are neither Lipschitz continuous nor differentiable in the classical sense. From now on, the expression "smooth manifold"—or, simply, the word "manifold"—means "finite-dimensional Hausdorff manifold of class C^1 without boundary." If M is a manifold, and $x \in M$, then T_xM, T_x^*M, TM, T^*M denote, respectively, the tangent and cotangent spaces of M at x, and the tangent and cotangent bundles of M. We start with several definitions.

A *finite family of state spaces* is a pair $(\mathcal{Q}, \mathcal{M})$ such that

FFSS1. $\mathcal{Q}$ is a finite set;

FFSS2. $\mathcal{M} = \{M_q\}_{q \in \mathcal{Q}}$ is a family of smooth manifolds, indexed by $\mathcal{Q}$.

If $(\mathcal{Q}, \mathcal{M})$ is a finite family of state spaces, then for each pair $(q, q') \in \mathcal{Q} \times \mathcal{Q}$ we use $\mathcal{M}_{q,q'}$ to denote the product $M_q \times M_{q'} \times r \times r$.

A *switching constraint* for a finite family of state spaces $(\mathcal{Q}, \mathcal{M})$ is a family $\mathcal{S} = \{S_{q,q'}\}_{(q,q') \in \mathcal{Q} \times \mathcal{Q}}$ such that $S_{q,q'}$ is a subset of $\mathcal{M}_{q,q'}$ for every pair $(q, q') \in \mathcal{Q} \times \mathcal{Q}$.

The following is the definition of "hybrid control system" that will be adopted for the purposes here. A *hybrid control system* is a 6-tuple

$$\Sigma = (\mathcal{Q}, \mathcal{M}, \mathcal{U}, f, \boldsymbol{U}, \mathcal{S})$$

such that

HCS1. $(\mathcal{Q}, \mathcal{M})$ is a finite family of state spaces;

HCS2. $\mathcal{U} = \{U_q\}_{q \in \mathcal{Q}}$ is a family of sets;

HCS3. $f = \{f_q\}_{q \in \mathcal{Q}}$ is a family such that f_q is, for each q, a partially defined map from $M_q \times U_q \times r$ to TM_q, having the property that $f_q(x, u, t)$ belongs to T_xM_q for every $(x, u, t) \in M_q \times U_q \times r$ for which $f_q(x, u, t)$ is defined;

HCS4. $\boldsymbol{\mathcal{U}} = \{U_q\}_{q\in\mathcal{Q}}$ is a family consisting, for each q, of a set U_q, each of whose members is a map $\eta : I_\eta \to U_q$ defined on some subinterval I_η of $\mathbb{R}$;

HCS5. $\mathcal{S} = \{S_{q,q'}\}_{(q,q')\in\mathcal{Q}\times\mathcal{Q}}$ is a switching constraint for $(\mathcal{Q},\mathcal{M})$.

The sets $S_{q,q'}$ are the *switching sets* of Σ, and are allowed to be empty. One should think of $S_{q,q'}$ as the set of all 4-tuples (x,x',t,t') such that $x \in M_q$, $x' \in M_{q'}$, and a switching (or "jump") from state $x \in M_q$ to state $x' \in M_{q'}$ is permitted at time t, with a resetting of the clock to time t'. Usually, one does not want to permit clock resetting, but for mathematical reasons it is better to allow it in principle, and exclude it, when desired, by just taking the switching sets $S_{q,q'}$ to consist only of points of the form (x,x',t,t).

The members of $\mathcal{Q}$ are called *locations*. The families $\mathcal{M}$, $\mathcal{U}$, are, respectively, the *family of state spaces* and the *family of control spaces* of Σ. For each q, the manifold M_q, the set U_q, the map f_q, and the set U_q are, respectively, the *state space*, the *control space*, the *dynamical law*, and the *class of admissible controls* at location q. Usually, $\mathcal{Q}$ will be the set of states of some finite automaton.

A *control* for a hybrid system Σ as above is a triple $\zeta = (\mathbf{q},\mathbf{I},\boldsymbol{\eta})$ such that

- $\mathbf{q} = (q_1,\ldots,q_\nu)$ is a finite sequence of locations;
- $\mathbf{I} = (I_1,\ldots,I_\nu)$ is a finite sequence of compact intervals;
- $\boldsymbol{\eta} = (\eta_1,\ldots,\eta_\nu)$ is a finite sequence such that η_j belongs to U_{q_j} and $I_{\eta_j} = I_j$ for $j = 1,\ldots,\nu$.

If $\zeta = (\mathbf{q},\mathbf{I},\boldsymbol{\eta})$ is a control, and $\mathbf{I} = (I_1,\ldots,I_\nu)$ for $j = 1,\ldots,\nu$, we use $\mathbf{q}(\zeta)$, $\mathbf{I}(\zeta)$, $\boldsymbol{\eta}(\zeta)$, $\nu(\zeta)$, to denote, respectively, the finite sequences $\mathbf{q}$, $\mathbf{I}$, $\boldsymbol{\eta}$, and the natural number ν. If $I_j = [t_j,\tau_j]$, we use $\mathbf{t}(\zeta)$, $\boldsymbol{\tau}(\zeta)$ to denote the sequences $(t_1,\ldots,t_\nu)$ and $(\tau_1,\ldots,\tau_\nu)$, and we let $a_\zeta = t_1$, $b_\zeta = \tau_\nu$. Then a_ζ, b_ζ, $\nu(\zeta)-1$, and $\mathbf{q}(\zeta)$ are, respectively, the *initial time*, the *terminal time*, the *number of switchings*, and the *switching strategy* of ζ.

If $\Sigma = (\mathcal{Q},\mathcal{M},\mathcal{U},f,\boldsymbol{\mathcal{U}},\mathcal{S})$ is a hybrid system as above, ζ is a control for Σ, and $\nu = \nu(\zeta)$, then a *pretrajectory* for ζ is a ν-tuple $\boldsymbol{\xi} = (\xi_1,\ldots,\xi_\nu)$ such that, if

$$\mathbf{I}(\zeta) = (I_1,\ldots,I_\nu)\,,\ I_j = [t_j,\tau_j],\ \mathbf{q}(\zeta) = (q_1,\ldots,q_\nu)\,,\ \boldsymbol{\eta}(\zeta) = (\eta_1,\ldots,\eta_\nu)\,,$$

then, for each $j \in \{1,\ldots,\nu\}$, ξ_j is an absolutely continuous map from I_j to the manifold M_{q_j}, having the property that $f_{q_j}(\xi_j(t),\eta_j(t),t)$ is defined and $\dot{\xi}_j(t) = f_{q_j}(\xi_j(t),\eta_j(t),t)$ for almost all $t \in I_j$.

If Σ is a hybrid system as above, a *pretrajectory-control pair* for Σ is a pair $(\boldsymbol{\xi},\zeta)$ such that ζ is a control for Σ and $\boldsymbol{\xi}$ is a pretrajectory of Σ for ζ.

We use $PTCP(\Sigma)$ to denote the set of all pretrajectory-control pairs of the system Σ.

An *endpoint constraint* for a finite family of state spaces $(\mathcal{Q},\mathcal{M})$ is a family $\mathcal{E} = \{E_{q,q'}\}_{(q,q')\in\mathcal{Q}\times\mathcal{Q}}$ of sets such that $E_{q,q'}$ is, for each $(q,q') \in \mathcal{Q}\times\mathcal{Q}$, a subset of $\mathcal{M}_{q,q'}$.

Notice that, mathematically, an endpoint constraint is exactly the same kind of object as a switching condition. This is why the part of the maximum principle that has to do with the switchings will have the same form as the transversality condition.

Let $\Sigma = (\mathcal{Q},\mathcal{M},\mathcal{U},f,\boldsymbol{\mathcal{U}},\mathcal{S})$ be a hybrid control system as in the previous definitions, and let $\Xi = (\boldsymbol{\xi},\zeta)$ belong to $PTCP(\Sigma)$. Let $\nu = \nu(\zeta)$, $\boldsymbol{\xi} = (\xi_1,\ldots,\xi_\nu)$, $\mathbf{q}(\zeta) = (q_1,\ldots,q_\nu)$, $\mathbf{t}(\zeta) = (t_1,\ldots,t_\nu)$, $\boldsymbol{\tau}(\zeta) = (\tau_1,\ldots,\tau_\nu)$, $\mathbf{I}(\zeta) = (I_1,\ldots,I_\nu)$, $\mathcal{S} = \{S_{q,q'}\}_{(q,q')\in\mathcal{Q}\times\mathcal{Q}}$. Then

- The *endpoint condition* of $\boldsymbol{\xi}$ (or of Ξ) is the 4-tuple

$$\partial\boldsymbol{\xi} \stackrel{\text{def}}{=} \partial\Xi \stackrel{\text{def}}{=} (\xi_\nu(b_\zeta), \xi_1(a_\zeta), b_\zeta, a_\zeta) \in \mathcal{M}_{q_\nu,q_1}\,. \tag{5}$$

- If $1 \leq j < \nu$, the "j-th jump" of $\boldsymbol{\xi}$ (or of Ξ) is the 4-tuple

$$\partial_j\boldsymbol{\xi} \stackrel{\text{def}}{=} \partial_j\Xi \stackrel{\text{def}}{=} (\xi_j(\tau_j), \xi_{j+1}(t_{j+1}), \tau_j, t_{j+1}) \in \mathcal{M}_{q_j,q_{j+1}}\,. \tag{6}$$

- If $\mathcal{E} = \{E_{q,q'}\}_{(q,q')\in\mathcal{Q}\times\mathcal{Q}}$ is an endpoint constraint for $(\mathcal{Q},\mathcal{M})$, we say that Ξ *satisfies the constraint* $\mathcal{E}$ if $\partial\Xi$ belongs to E_{q_ν,q_1}.

- We say that $\boldsymbol{\xi}$ (or Ξ) *satisfies the switching conditions for* Σ if $\partial_j\Xi$ belongs to $S_{q_j,q_{j+1}}$ whenever $j \in \{1,\ldots,\nu-1\}$.

If $\Sigma = (\mathcal{Q},\mathcal{M},\mathcal{U},f,\boldsymbol{\mathcal{U}},\mathcal{S})$ is a hybrid system as above, then

- we say that a pretrajectory $\boldsymbol{\xi}$ of Σ is a *trajectory* of Σ if $\boldsymbol{\xi}$ satisfies the switching conditions for Σ;

- we use $TCP(\Sigma)$ to denote the set of all trajectory-control pairs of Σ (i.e., the set of all $\Xi = (\boldsymbol{\xi},\zeta) \in PTCP(\Sigma)$ such that $\boldsymbol{\xi}$ is a trajectory of Σ), and $TCP(\Sigma;\mathcal{E})$ to denote the set of all $\Xi \in TCP(\Sigma)$ that satisfy the endpoint constraint $\mathcal{E}$.

If Σ is a hybrid system as above, then a *Lagrangian* for Σ is a family $L = \{L_q\}_{q\in\mathcal{Q}}$ such that

- L_q is, for each $q \in \mathcal{Q}$, a partially defined real-valued function on the product $M_q \times U_q \times r$,

- whenever $q \in \mathcal{Q}$, $\eta \in U_q$ has domain $[\alpha,\beta]$, and $\xi : [\alpha,\beta] \to M_q$ is an absolutely continuous solution of $\dot{\xi}(t) = f_q(\xi(t),\eta(t),t)$ a.e., it follows that the function $[\alpha,\beta] \ni t \to L(\xi(t),\eta(t),t)$ is defined for almost every t, and is integrable.

A *switching cost function* for Σ is a family $\Phi = \{\Phi_{q,q'}\}_{(q,q')\in\mathcal{Q}\times\mathcal{Q}}$ such that each $\Phi_{q,q'}$ is an extended real-valued function on $S_{q,q'}$ that never takes the value $-\infty$.

An *endpoint cost function* for Σ is a family $\varphi = \{\varphi_{q,q'}\}_{(q,q')\in\mathcal{Q}\times\mathcal{Q}}$ such that each $\varphi_{q,q'}$ is an extended real-valued function on $\mathcal{M}_{q,q'}$ that never takes the value $-\infty$.

If $L = \{L_q\}_{q\in\mathcal{Q}}$ is a Lagrangian for the hybrid control system Σ, then we can define the corresponding *Lagrangian cost functional* $C_L : TCP(\Sigma) \to r$, by letting

$$C_L(\boldsymbol{\xi},\zeta) = \sum_{j=1}^{\nu} \int_{I_j} L_{q_j}(\xi_j(t),\eta_j(t),t)\,dt\,, \tag{7}$$

where $\nu = \nu(\zeta)$, $\mathbf{I}(\zeta) = (I_1,\ldots,I_\nu)$, $\mathbf{q}(\zeta) = (q_1,\ldots,q_\nu)$, $\boldsymbol{\eta}(\zeta) = (\eta_1,\ldots,\eta_\nu)$, and $\boldsymbol{\xi} = (\xi_1,\ldots,\xi_\nu)$.

If Φ is a switching cost function for Σ, and φ is an endpoint cost function, then we associate with Φ and φ the functional $\hat{C}_{\Phi,\varphi} : TCP(\Sigma) \to r \cup \{+\infty\}$ that assigns to each $\Xi = (\boldsymbol{\xi},\zeta) \in TCP(\Sigma)$ the number

$$\hat{C}_{\Phi,\varphi}(\boldsymbol{\xi},\zeta) = \varphi_{q_\nu,q_1}(\partial\Xi) + \sum_{j=1}^{\nu-1} \Phi_{q_j,q_{j+1}}(\partial_j\Xi)\,, \tag{8}$$

where $\nu = \nu(\zeta)$, and $(q_1, \ldots, q_\nu)$ is the switching strategy of ζ.

A *hybrid Bolza cost functional* for Σ is an extended real-valued functional $\mathbb{C} : TCP(\Sigma) \to r \cup \{+\infty\}$ such that $\mathbb{C} = C_L + \hat{C}_{\Phi,\varphi}$ for some L, Φ, φ that are, respectively, a Lagrangian, a switching cost function, and an endpoint cost function for Σ.

Given a hybrid control system Σ, a Bolza cost functional $\mathbb{C}$ for Σ, and an endpoint constraint $\mathcal{E}$, we will consider the *optimal control problem* $\mathcal{P}(\Sigma, \mathbb{C}, \mathcal{E})$, whose objective is to minimize $\mathbb{C}(\boldsymbol{\xi}, \zeta)$ in the class $TCP(\Sigma; \mathcal{E})$. We observe that the endpoint constraint sets $E_{q,q'}$ could all be of the special form $E^0_{q,q'} \times \{b\} \times \{a\}$, where a, b are fixed real numbers, independent of q, q', and each $E^0_{q,q'}$ is a subset of $M_q \times M_{q'}$. In that special case, all the members $\Xi = (\boldsymbol{\xi}, \zeta)$ of $TCP(\Sigma; \mathcal{E})$ satisfy $a_\zeta = a$, $b_\zeta = b$, so we have a *problem with fixed initial and terminal times.* In addition, the switching sets $S_{q,q'}$ could be of the form $S_{q,q'} = S^0_{q,q'} \times \{\bar{t}_{q,q'}\} \times \{\bar{t}_{q,q'}\}$, where $S^0_{q,q'} \subseteq M_q \times M_{q'}$ and the $\bar{t}_{q,q'}$ are fixed real numbers, in which case we would be dealing with a *problem with fixed switching times and no clock resetting.*

7.1 The general form of the maximum principle

Let us assume that

A1. $\Sigma = (\mathcal{Q}, \mathcal{M}, \mathcal{U}, f, \boldsymbol{\mathcal{U}}, \mathcal{S})$ is a hybrid control system;

A2. $\mathbb{C} = C_L + \hat{C}_{\Phi,\varphi}$ is a hybrid Bolza cost functional for Σ;

A3. $\mathcal{E}$ is an endpoint constraint for $(\mathcal{Q}, \mathcal{M})$;

A4. $\Xi^\#$ (the "reference trajectory-control pair") belongs to $TCP(\Sigma; \mathcal{E})$, and

$$\begin{array}{rclcrcl} \Xi^\# & = & (\boldsymbol{\xi}^\#, \zeta^\#), & & \boldsymbol{\xi}^\# & = & (\xi_1^\#, \ldots, \xi_{\nu^\#}^\#), \\ \zeta^\# & = & (\mathbf{q}^\#, \mathbf{I}^\#, \boldsymbol{\eta}^\#), & & \mathbf{q}^\# & = & (q_1^\#, \ldots, q_{\nu^\#}^\#), \\ \mathbf{I}^\# & = & (I_1^\#, \ldots, I_{\nu^\#}^\#), & & \boldsymbol{\eta}^\# & = & (\eta_1^\#, \ldots, \eta_{\nu^\#}^\#). \end{array}$$

The *maximum principle* gives a necessary condition for $\Xi^\#$ to be a solution of $\mathcal{P}(\Sigma, \mathbb{C}, \mathcal{E})$. The result only depends on comparing trajectories with the same switching strategy, and does not require the candidate arc $\Xi^\#$ to be a true solution. Moreover, even within the class of arcs corresponding to a fixed switching strategy, only arcs that are close to $\Xi^\#$ are compared with $\Xi^\#$. So we introduce the following definition.

A *local solution* of a problem $\mathcal{P}(\Sigma, \mathbb{C}, \mathcal{E})$ is a trajectory-control pair $\Xi^\# = (\boldsymbol{\xi}^\#, \zeta^\#) = (\xi_1^\#, \ldots, \xi_{\nu^\#}^\#, \zeta^\#)$ such that there exist neighborhoods $\mathcal{N}_1, \ldots, \mathcal{N}_{\nu^\#}$ of the graphs of $\xi_1^\#, \ldots, \xi_{\nu^\#}^\#$ in $M_{q_1} \times r, \ldots, M_{q_\nu^\#} \times r$ having the property that $\Xi^\#$ minimizes the cost $\mathbb{C}(\Xi)$ in the class of all the trajectory-control pairs $\Xi = (\boldsymbol{\xi}, \zeta) = (\xi_1, \ldots, \xi_\nu, \zeta) \in TCP(\Sigma, \mathcal{E})$ such that $\mathbf{q}(\zeta) = \mathbf{q}(\zeta^\#)$ (so that, in particular, $\nu = \nu^\#$) and the graph $G(\xi_j)$ of ξ_j is contained in $\mathcal{N}_j$ for $j = 1, \ldots, \nu^\#$. (Here the "graph" of ξ_j is the set

$$G(\xi_j) \stackrel{\text{def}}{=} \{(\xi_j(t), t) : t \in \text{Domain}(\xi_j)\}, \tag{9}$$

so $G(\xi_j) \subseteq M_{q_j} \times r$.)

We now present the maximum principle for hybrid systems as a true "principle," that is, a not very precise mathematical statement that can be rendered precise in various ways, giving rise to different "versions" of the principle. Two such versions—both completely precise and rigorous—are stated in the papers [20], [21], and [22].

The maximum principle. *Assume that A1-A4 hold, and $\Xi^{\#}$ is a local solution of $\mathcal{P}(\Sigma, \mathbb{C}, \mathcal{E})$. Then there exists an adjoint pair $(\boldsymbol{\psi}, \psi_0)$ along $\Xi^{\#}$ that satisfies the weak Hamiltonian maximization, nontriviality, and transversality conditions for $\mathcal{P}(\Sigma, \mathbb{C}, \mathcal{E})$ along $\Xi^{\#}$.*

To turn the above statement into a theorem, we have to specify technical assumptions on the 12-tuple of data $(\mathcal{Q}, \mathcal{M}, \mathcal{U}, f, \boldsymbol{U}, \mathcal{S}, L, \Phi, \varphi, \mathcal{E}, \boldsymbol{\xi}^{\#}, \zeta^{\#})$, and assign a precise meaning to the notions of "adjoint pair," "weak Hamiltonian maximization," "nontriviality," and "transversality." This is done in the above papers.

List of Papers (Journals, Book Chapters, Conference Proceedings) on this Project, Submitted or Appeared During the Grant Period

1. (with C. Darken, C., M. Donahue, and L. Gurvits) "Rates of convex approximation in non-Hilbert spaces," *Constructive Approximation* **13**(1997): 187-220

2. Y. Yang,, E. Sontag, H.J. Sussmann, "Global stabilization of linear discrete-time systems with bounded feedback," *Systems and Control Letters*, **30** (1997): 273-281.

3. P. Koiran, E.D. Sontag, "Neural networks with quadratic VC dimension," *J. Comp. Syst. Sci.* **54**(1997): 190-198.

4. E.D. Sontag, H.J. Sussmann, "Complete controllability of continuous-time recurrent neural networks," *Systems and Control Letters* **30**(1997): 177-183.

5. E.D. Sontag, "Shattering all sets of k points in 'general position' requires $(k-1)/2$ parameters," *Neural Computation* **9**(1997): 337-348.

6. R. Koplon, E.D. Sontag, "Using Fourier-neural recurrent networks to fit sequential input/output data," *Neurocomputing* **15**(1997): 225-248.

7. P. Koiran, E.D. Sontag, "Vapnik-Chervonenkis dimension of recurrent neural networks," *Discrete Applied Math.* **86**(1998): 63-79.

8. E.D. Sontag, "A learning result for continuous-time recurrent neural networks," *Systems and Control Letters* **34**(1998): 151-158

9. Y. Qiao, E.D. Sontag, "Further results on controllability of recurrent neural networks," *Systems and Control Letters* **36**(1999): 121-129.

10. W. Maass, E.D. Sontag, "Analog neural nets with gaussian or other common noise distributions cannot recognize arbitrary regular languages," *Neural Computation* **11**(1999): 771-782.

11. B. Dasgupta, E.D. Sontag, "A polynomial-time algorithm for checking equivalence under certain semiring congruences motivated by the state-space isomorphism problem for hybrid systems," *Theoret. Comp. Sci*, to appear.

12. W. Maass, E.D. Sontag, "Neural systems as nonlinear filters," *Neural Computation* **12**(2000): 1743-1772.

13. X. Bao and Z. Lin, E.D. Sontag, "Finite gain stabilization of discrete-time linear systems subject to actuator saturation," *Automatica* **36**(2000): 269-277.

14. E.D. Sontag, "Structure and stability of certain chemical networks and applications to the kinetic proofreading model of T-cell receptor signal transduction," *IEEE Trans. Autom. Control*, to appear.

15. D. Ocone, P. Kuusela, E.D. Sontag, "Learning-complexity dimensions for a continuous-time control system," submitted to *SIAM J. Control and Optimization.*

16. Y. Ledyaev, E.D. Sontag, "A notion of discontinuous feedback," in *Control Using Logic-Based Switching* (A.S. Morse, ed.), pp. 97-103, Springer-Verlag, London, 1997.

17. E.D. Sontag, "Recurrent neural networks: Some systems-theoretic aspects," in *Dealing with Complexity: a Neural Network Approach* (M. Karny, K. Warwick, and V. Kurkova, eds.), Springer-Verlag, London, 1997, pp. 1-12.

18. E.D. Sontag, "VC dimension of neural networks," in *Neural Networks and Machine Learning* (C.M. Bishop, ed.), Springer-Verlag, Berlin, 1998, pp. 69-95.

19. P. Kuusela, D. Ocone, E.D. Sontag, "On the VC dimension of continuous-time linear control systems," in *Proc. 32nd Annual Conf. on Information Sciences and Systems (CISS 98)*, Princeton, NJ, 1998, pp. 795-800.

20. H.J. Sussmann, "Set-valued differentials and the hybrid Maximum Principle," *Proc. 39th IEEE Conference on Decision and Control*, Sydney, Australia, Dec. 2000.

21. H.J. Sussmann, "A Maximum Principle for hybrid optimal control problems," *Proc. 38th IEEE Conference on Decision and Control*, Phoenix, AZ, Dec. 1999), pp. 425-430.

22. H.J. Sussmann, "A nonsmooth hybrid Maximum Principle," in *Stability and Stabilization of Nonlinear Systems* (D. Aeyels, F. Lamnabhi-Lagarrigue, and A.J. van der Schaft, Eds.), Lecture Notes in Control and Information Sciences, no. 246, Springer-Verlag, pp. 325-354.

23. Y. Qiao, E.D. Sontag, "Remarks on controllability of recurrent neural networks," in *Proc. IEEE Conf. Decision and Control, Tampa, Dec. 1998*, IEEE Publications, 1998, pp. 501-506.

24. X. Bao, Z. Lin, E.D. Sontag, "Some new results on finite gain l_p stabilization of discrete-time linear systems subject to actuator saturation," in *Proc. IEEE Conf. Decision and Control, Tampa, Dec. 1998*, IEEE Publications, 1998, pp. 4628-4629.

25. B. Dasgupta, E.D. Sontag, "A polynomial-time algorithm for an equivalence problem which arises in hybrid systems theory," in *Proc. IEEE Conf. Decision and Control, Tampa, Dec. 1998*, IEEE Publications, 1998, pp. 1629-1634.

26. W. Maass, E.D. Sontag, "A precise characterization of the class of languages recognized by neural nets under gaussian and other common noise distributions," in *Advances in Neural Information Processing Systems 11* (NIPS98), MIT Press, Cambridge, 1999, 281-287.

27. T. Natschläger, W Maass, E.D. Sontag, A. Zador, "Processing of time series by neural circuits with biologically realistic synaptic dynamics," in *Advances in Neural Information Processing Systems 13* (NIPS2000), MIT Press, Cambridge, 2001, to appear.